TeX: A Comprehensive Look

CS846 Electronic Publishing
University of Waterloo

Prasanna Kumar

What is TeX?

What is TeX

•TEX is a typesetting system designed by Donald Knuth.

•It is very popular in academia, especially in Mathematics, Computer Science, Engineering and Statistics.

•It is a free software.

•Many publishers especially those with a Engineering and Mathematics focus have adopted it as their preferred format.

TEX

What is in the name?

- English word "technology" comes from a greek root with the letters tau,epsilon,chi.

- So the name TEX comes from the upper cases of those greek letters (T E X).

- So, the last letter isn't pronounced as X but as "ch".

- The displaced E is to distinguish from other names and also to denote that TeX is about typesetting.

A Brief History of TeX

- TeX was a project started in 1978 by Donald E Knuth.

- Revising second volume of "Art of Computer Programming".

- Shocked at the poor quality of digital typesetting system of the publisher.

- Estimated 6 months, took about a decade

- He had to handle routine typesetting as well as make page formatting flexible enough to allow different output styles. Moreover, had to tell the computer how to typeset formulas, footnotes, tables, etc.

A Brief History of TeX

- TeX82: rewritten from scratch.

- In 1989, Knuth made one final set of changes (primarily moving from a 7 bit to 8 bit character sets system)

- Source code is essentially Public Domain and is considered to be permissive free software.

- 1985: LaTeX by Leslie Lamport gave a huge boost in popularity.

- ConTeXt also boosted popularity of Tex in 1990.

Qualities of TeX that make it useful

There are major feature of TeX systems. They can be put into four areas listed below (Source: https://ctan.org/tex)

- Output Quality

- Superior Engineering

- Freedom

- Popularity

Features of TeX
Output Quality

The first concern for any document: it should be easy and pleasing to read. Therefore, our biggest concern with any such formatter or editor should be : "How good is the output".

- Great Output: TeX produces really clean and well structured outputs. Example: https://ctan.org/tex/rogers.pdf

- Typesetting: It deals with mathematical formulas as well as text. For example, writing $\sqrt{(a^2 + b^2)}$ in a proper easy to read way is very difficult in a word processor. However, in TeX, we simply write \sqrt{ a^{2} + b ^{2}}. We don't have to think about how to get the root symbol to cover the entire expression.

Recursive definitions are familiar in mathematics. For instance, the function f defined by

$$f(0) = 1,$$
$$f(1) = 1,$$
$$f(x + 2) = f(x + 1) + f(x),$$

gives the Fibonacci sequence: 1, 1, 2, 3, 5, 8, 13, ... (The study of *difference equations* concerns the problem of going from recursive definitions to algebraic definitions. The Fibonacci sequence is give by the algebraic definition

$$f(x) = \frac{\sqrt{5}}{5}\left(\frac{1 + \sqrt{5}}{2}\right)^{x+1} - \frac{\sqrt{5}}{5}\left(\frac{1 - \sqrt{5}}{2}\right)^{x+1}.)$$

The *primitive recursive functions* are an example of a broad and interesting class of functions that cam be obtained by such a formal characterization.

Definition The class of *primitive recursive functions* is the smallest class $\mathcal{C}$ (i.e., intersection of all classes $\mathcal{C}$) of functions such that

i. All *constant functions*, $\lambda x_1 x_2 \cdots x_k[m]$ are in $\mathcal{C}$, $1 \le k$, $0 \le m$;

ii. The *successor function*, $\lambda x[x + 1]$, is in $\mathcal{C}$;

iii. All *identity functions*, $\lambda x_1 \cdots x_k[x_i]$ are in $\mathcal{C}$, $1 \le i \le k$;

iv. If f is a function of k variables in $\mathcal{C}$, and $g_1, g_2, \ldots, g_k$ are (each) functions of m variables in $\mathcal{C}$, then the function $\lambda x_1 \cdots x_m[f(g_1(x_1, \ldots, x_m), \ldots, g_k(x_1, \ldots, x_m))]$ is in $\mathcal{C}$, $1 \le k, m$;

v. If h is a function of $k + 1$ variables in $\mathcal{C}$, and g is a function of $k - 1$ variables in $\mathcal{C}$, then the unique function f of k variables satisfying

$$f(0, x_2, \ldots, x_k) = g(x_2, \ldots, x_k),$$
$$f(y + 1, x_2, \ldots, x_k) = h(y, f(y, x_2, \ldots, x_k), x_2, \ldots, x_k)$$

is in $\mathcal{C}$, $1 \le k$. (For (v), "function of zero variables in $\mathcal{C}$" is taken to mean a fixed integer.)

Features of TeX
Superior Engineering

- Fast, Easy on Memory and Disk Space

- Stable: TeX's Designer has frozen the Central engine

- Stable but not rigid

- Input is plain text: Portable, compact, text can be automatically generated due to this plain text input. This also means that articles containing even complicated formulas can be easily transmitted electronically.

- Output can be anything: outputting is separate from Typesetting. Therefore, the Tex's engine results can be output as many different formats.

Features of TeX
Freedom

- Free: The source of the main TeX engine is open. FSF uses it for their documents.

- TeX runs on virtually all platforms. And it is platform independent. So, we may type the source file on a machine X and then transmit to a co-author who can edit on their machine Y without any problem even if the two machines are running two different ecosystems.
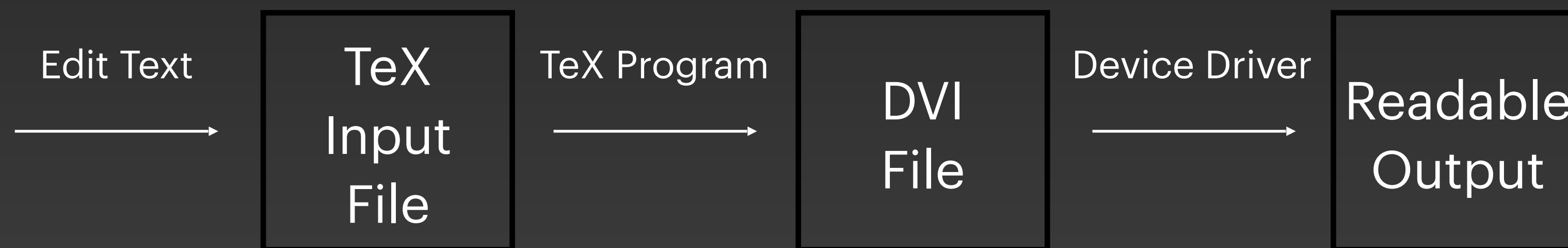
Features of TeX
Standard

- Popularity wise: Many publishers involved with technical materials and most scientific publications use TeX.

How do we use TeX?

- TeX is a typesetting language.

- Usually, we visually format our text.

- Here, we write our text in a plain text file intertwined with TeX commands.

- Then we run TeX to produce an output which can be in many different formats.

- Typesetting commands are preceded by "\" and arguments go into {}.

How TeX works

- Step 1: Type the file that TEX reads. This is called the TeX file or input file.

- Step 2: TeX program reads the file from Step 1 and produces a DVI file which isn't readable by humans

- Step 3: This DVI file will be read by a device driver which will produce the output file.

- Why DVI?

- TeX can be run in two modes: Interactive or Batch

```
Edit Text          TeX Program              Device Driver
   ────────▶  ┌──────────┐  ────────▶  ┌──────────┐  ────────▶  ┌──────────┐
              │   TeX    │              │   DVI    │              │ Readable │
              │  Input   │              │   File   │              │  Output  │
              │   File   │              │          │              │          │
              └──────────┘              └──────────┘              └──────────┘
```

Examples of TeX Commands

- Here is my first \TeX\ sentence.
  \bye

  Here is my first TeX sentence.

- The quadratic formula is $-b \pm \sqrt{b^2 - 4ac} \over 2a$

  The quadratic formula is $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

TeX and Friends

Notable extensions and versions of TeX

- Knuth froze TeX and has declared that the program will not change in any substantial way.

- However, the source code is available in public domain and can be extended.

- Many Versions and Extensions have been developed over the years. Some are general purpose while others have a particular purpose.

- We will discuss some of these in the following slides.

Versions of Tex

- Strictly speaking, there is only one definitive version of TeX that Knuth developed and maintains.

- The symbol of TeX is a trademark of AMS

- However, Knuth allowed people to add functionality and changes that he didn't but with one condition: the changed program can not be called TeX.

- So, when we say "versions" of TeX, we are referring to the adaptations and derivatives of TeX.

- Some aspects of TeX like file format started showing its age as time progressed and developers wanted to build upon the strengths of TeX.

TeX the program and TeX the programming language

- TeX doesn't have a GUI.

- We have to write specific instructions to guide it through the typesetting process.

- So our input file has both the text that we want to type and the instructions for the typesetting process.

- The typesetting instructions are written in a language designed by Knuth.

- This PL gives the incredible flexibility and power to TeX.

From TeX to infinity

- Technology has evolved a lot since TeX was written: computers are faster now, they have more memory, new font technologies developed such as Type 1, TrueType and OpenType, file output formats have evolved (rise of PDF) and Unicode has become dominant.

- So, a lot of developers have developed new versions which don't just add these features but extend and adapt the language.

- These new TeX-based engines provide more power to users in terms of programming capabilities and flexibility.

- They all have different names such as pdfTeX, LuaTeX.

TeX Macros

- There is one problem that arises in the previous versions of TeX that we discussed.

- Since each version has their own additional set of primatives, these may not be compatible with each other.

- Of course, most of the primitives will be supported by all but some won't and these can cause issues.

- So, what is the solution?

- TeX Macros: they are new commands built by combining primitives or/and other macros.

LaTeX

- LaTeX is a set of macros built on top of TeX and was designed by Leslie Lamport.
- Major focus is given to document structure and the logical markup of tex and automatic numbering and cross-referencing. (Source: http://tug.ctan.org/info/mil/mil.pdf)
- We put focus on content instead of formatting.
- It is simply put a large collection of sophisticated macros to help typeset books, journal papers, etc..

What is LaTex?
Source: WikiBooks

- It is a set of macros for TeX designed by Leslie Lamport.

- It is a Document Preparation System.

- In TeX, author has to take care of both : Typographical and logical markup language.

- In LaTeX, Leslie aimed to separate the two.

- A typesetter can take care of making a template and the author just focuses on logical markup without knowing anything about typesetting.

- It was designed to be extensible. (We can plug-in additional, more specialist, macro packages).

pdfLaTeX, LuaLaTeX, XeLaTeX
Source: overleaf.com

- When we say we are typesetting a document using LaTeX: we are referring to are using the LaTeX macro package on some TeX engine.

- "pdfLaTeX means using the LaTeX macro package with the pdfTeX engine

- XeLaTeX means using the LaTeX macro package with the XeTeX engine

- LuaLaTeX means using the LaTeX macro package with the LuaTeX engine

LaTeX Examples

- Let's take a look at some examples of parts of documents created using LaTeX.

- We use Overleaf (will be discussed later) to run some useful LaTeX commands

- We will see how simple LaTeX is to write for producing well structured document.

- So, let's go ahead :

Document Page 1

```latex
\documentclass{article}

\usepackage[utf8]{inputenc}


\title{Sample for SE846}

\author{Prasanna Kumar}

\date{July 2021}


\begin{document}

\maketitle
```


Sample for SE846

Prasanna Kumar

July 2021

Document Page 2: Item

```
\section{Introduction}

\begin{itemize}

    \item Item 1

    \item Item 2

    \item Item 3

    \item Item 4

\end{itemize}
```


Check this header

1 Introduction

- Item 1
- Item 2
- Item 3
- Item 4

Document Page 2: Table

```
\begin{table}[ht!]

\centering

\begin{tabular}{c|c|c|c|c}

 & X & O & X &  \\ \hline

X& X & O & X &X  \\ \hline

 & X & O & X &  \\

\end{tabular}

\caption{Table Example}

\end{table}
```


Table 1: Table Example

Document Page 2: Section (Defining Environments)

\begin{answer}

Here we put a text

\end{answer}

\end{document}

Check this header

1 Introduction

- Item 1
- Item 2
- Item 3
- Item 4

X	O	X
X	X	O	X	X
X	O	X

Table 1: Table Example

Answer: Here we put a text

LaTeX
Interview with Leslie Lamport

- Leslie needed to write a set of macros for the book. So, he decided to make an extra effort and make macro package that can be used by other people as well. That was the origin of LATEX.

- An interesting quote about whether being free helped LaTeX : "I think I made more money by giving the software away and selling the book than I would have by trying to sell the software. I don't think TEX and LATEX would have become popular had they not been free. Indeed, I think most users would have been happier with Scribe".

- Leslie states that he feels that the software is easy to use only for a small group of people.

LaTeX
Interview with Leslie Lamport

- In the interview, interestingly, Leslie believes that TeX will not last very long as people in today's world are too used to softwares like Word and won't use an arcane system such as TeX.

- He states that people should focus more on content and less on formatting when dealing with LaTeX.

- He believes that the reason there is no fully featured WYSIWYG system that can match the efficiency of TeX because there is no demand for such a software.

ConTeXt

- ConTeXt: it is a general purpose document processor and just like LaTeX, it is derived from TeX.

- This macro package started as a production tool for a publishing company (Pragma).

- LaTeX hides the typographical details and allows user to worry about to content more than format.

- However, ConTeXt takes a complementary approach.

ConTeXt

- It gives user a lot of control over the formatting.

- It has a unified design and avoids package clashes that can happen in LaTeX.

- To make a ConTeXt: we take a plain text file and the compile using the context script (output is PDF).

ConTeXt: Advantages over LaTeX

- LaTeX was created and designed with the idea of separating user content and formatting such that all the author has to do is write the content for the document and the style that is used is created by someone else.

- ConTeXt is similar in approach but the end user does all the "layout". So it provides a vast amount of flexibility for layout.

- Since it was designed later, it was made for faster computers which makes it slower but easier in terms of programming.

- Use of LuaTeX.

- ConTeX is a more powerful, integrated and flexible macropackage.

A short example for ConTeXt

- \setuphead[section][color=red]

  \starttext

  \section{Hello}

  World!

  \stoptext

BibTex

- It is a software that is used for managing references and formatting a list of references.

- Often, it is used with LaTeX.

- The primary purpose of this software is to allow us to have an easy way to cite sources.

- Main advantage is that we can put the bibliographic information separately.

- It was created by Oren Patashnik and Leslie Lamport in 1985.

Bibliography: just a list of `\bibitems`

\begin{thebibliography}{9}

\bibitem{texbook}

Donald E. Knuth (1986) \emph{The \TeX{} Book}, Addison-Wesley Professional.

\bibitem{lamport94}

Leslie Lamport (1994) \emph{\LaTeX: a document preparation system}, Addison

Wesley, Massachusetts, 2nd ed.

\end{thebibliography}

Then,

\LaTeX{} \cite{lamport94} is a set of macros built atop \TeX{} \cite{texbook}.

LaTeX [1] is a set of macros built atop TeX [2].

References

[1] Leslie Lamport (1994) *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd ed.

[2] Donald E. Knuth (1986) *The TeX Book*, Addison-Wesley Professional.

Inconveniences with manually preparing the `thebibliography` list

- Need to accurately format each bibitem

- Manual sorting of items needed if we want to reorganise the list in alphabetical order

Enter BibTeX

- We maintain a bibliography database. Example: refs.bib

- *format-independent* information about our references

- Example: @book{texbook,

  author = {Donald E. Knuth},

  year = {1986},

  title = {The \TeX{} Book},

  publisher = {Addison-Wesley Professional}

  }

- Now, we can use \cite to use the references.

- \bibliographystyle{} : we use this to choose the style of referencing for our document

- \bibliography{..}: to point to the .bib file we want to use

Example

Report.bib

```
@book{SS14,
  title = {Understanding Machine Learning: From Theory to Algorithms},
  author = {Shai Shalev-Shwartz and Shai Ben-David},
  publisher = {Cambridge University Press},
  year = {2014},
}

@article{Block62,
  title        = {The perceptron: A model for brain functioning},
  author       = {H. D. Block},
  journal      = {Reviews of Modern Physics},
  volume       = {34},
  number       = {1},
  pages        = {123--135},
  year         = {1962},
}
```

LaTeX

\bibliographystyle{aaai}
\bibliography{report}

Result.pdf

References

Alexander Gro, J. F., and Schwenker, F. 2008. Learning to play tetris applying reinforcement learning methods. *DBLP:conf/esann/GrossFS08*.

Alexander Gro, J. F., and Schwenker, F. 2013. Torchcraft: a library for machine learning research on real-time strategy games. *arXiv preprint arXiv: 1611.00625v1*.

Andre Brandao, Pedro Pires, P. G. Reinforcement learning and neuroevolution in flappy bird game.

Dino S. Ratcliffe, Sam Devlin, U. K. L. C. Clyde: A deep reinforcement learning doom playing agent.

BibLATEX – Sophisticated Bibliographies in LATEX

- This is a reimplementation of the previous mentioned software.

- It uses its own backend program called biber to read and process bilbiographic data.

- It has a lot of features such as customisable bibliography labels, remote data sources, subdivided and filtered bibliographies, etc..

- The package incorporates features of packages such as multibib, splitbib, mcite, chapterbib, etc.

How to use: BibLaTeX

- Existing .bib files will work

- In preamble, we will specify that we want to use this package and furthermore, we will have to specify the .bib files to use.

- The commands that we will use are : \usepackage{biblatex} and \addbibresource{}

- In BibTeX, we specified the style of referencing using \bibliographystyle command

- In BibLaTeX, we don't have to do this. Instead we can specify the style in the usepackage command itself. For example, \usepackage[style=numeric]{biblatex}.

- We can also specify different styles for citations and different style for the bibliography. \usepackage[citestyle=alphabetic,bibstyle=authortitle]{biblatex}

Styles in BibLaTeX

- The numeric style:

  This is the numeric style[1].

  References

  [1]   Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

- The alphabetic style:

  This is the alphabetic style[GMS93].

  References

  [GMS93]   Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

Styles in BibLaTeX

- The reading style:

- The author year style:

More commands in BibLaTeX

- We can sort the references by giving the sorting preference in the \usepackage command. For example, \usepackage[style=authoryear,sorting=ynt]{biblatex}. Note: ynt—sorts entries by year, name, title;

- Overhaul of the citation command: we can add prenote and postnote. Prenote is a word or expression inserted at the beginning and postnote is inserted at the end of the citation. Example, \cite[see][page 12]{latexcompanion}

- Single [] means postnote. For empty postnote, [name][] should be used.

- Subdividing the bibliography is possible using the source type.

Example for Subdividing

- \printbibheading

- \printbibliography[type=book,heading=subbibliography,title={Book Sources}]

- \printbibliography[nottype=book,heading=subbibliography,title={Other Sources}]

References

Book Sources

[GMS93]   Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

Other Sources

[Ein05]   Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[Knu]   Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/~uno/abcde.html.

pdfTeX

- It is an extension of TeX that can generate PDF documents (as well as DVI output).

- All current distribution of TeX include pdfTeX.

- Written by Han The Thanh

- There are two main extensions: margin kerning and composing with font expansion.

- It is included in most modern distributions of LaTeX and ConTeX.

- It gives a direct access to PDF specific feature which are absent in TeX (such as hyperlinks).

LuaTeX

- It is an extended version of pdfTeX using Lua as an embedded scripting language.

- Its main objective is to provide an open and configurable variant of TeX and offer backward compatibility.

- It was developed by Taco Hoekwater, Hartmut Henkel, Hans Hagen.

- It provides a version of TeX where all internals are open and accessible to Lua.

- It supports OpenType fonts with external modules.

LuaTeX

- LuaTeX has a powerful scripting language (namely, Lua) built in.

- This means we can use Lua to load plugins into LuaTeX

- This further enables high levels of automation, integration into existing software systems or workflows and leveraging specialist software for data, text or graphics processing.

- Applications outside of Mathematics Domain: commercial PDF document production

- Example: speedata publisher which uses LuaTeX purely as the PDF-generation engine within its XML-based workflow—it does not use LaTeX at all.

XeTeX

- It is a unicode TeX engine.

- Loads system fonts directly using the HarfBuzz Library which is built in

- \font primitive is extended in order to achieve this.

- Written by Jonathan Kew

- It was originally made for MacOS but now supports all platform.

- It doesn't directly produce a pdf. Instead, it first produces a xdv file which is then converted to PDF.

- It works with both LaTeX and ConTeXt.

- To invoke the counterpart in LaTeX, we invoke xelatex

- Usually used with the fontspec package.

XeTeX Example

- Times New Roman font using XeLaTeX Compiler (Source: Overleaf)

```
\documentclass[12pt]{article}
\usepackage{fontspec}

\setmainfont{Times New Roman}
\title{Sample font document}
\author{Overleaf}
\date{\today}

\begin{document}
\maketitle
```

This is an \textit{example} of a document compiled
with \textbf{XeLaTeX}. LuaLaTeX should also work too.

```
\end{document}
```

MusiXTeX

- MusiXTeX: It is a TeX derivative. It is a collection of music engraving macros that allows music typesetting in TeX. MuTeX and MusicTeX were the first to aim to achieve this goal. This new module requires a three-pass compilation: TeX, musixflx and TeX again.

Thank You